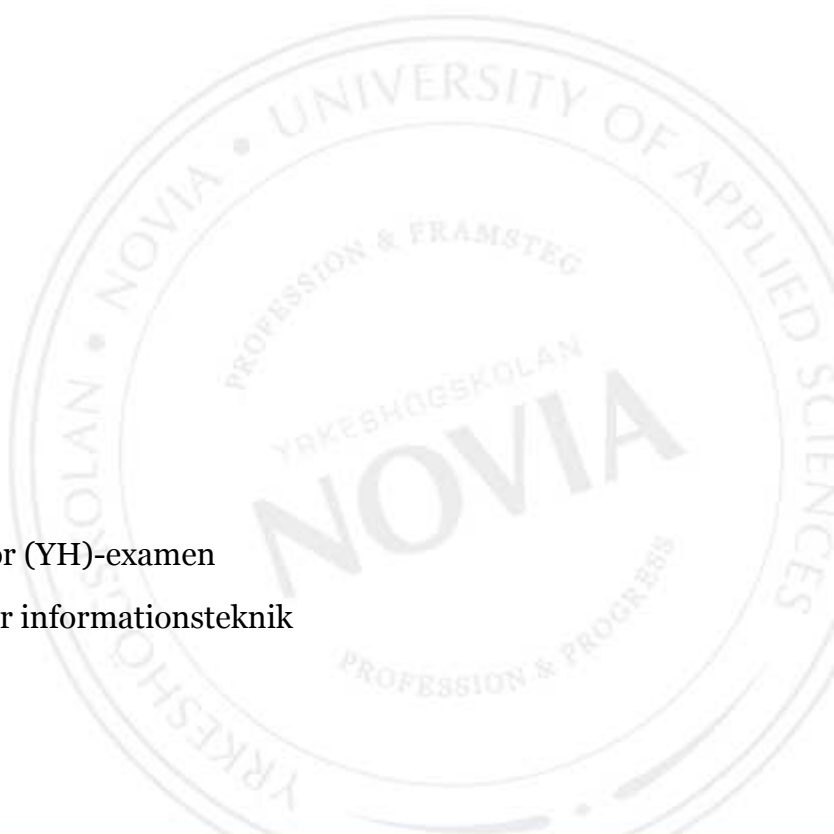


# **WEBBASERAD BOKNINGSKALENDER OCH DATORSTYRNING AV VÄRMESYSTEM**

Christoffer Smeds

Examensarbete för Ingenjör (YH)-examen  
Utbildningsprogrammet för informationsteknik  
Vasa 2011



## EXAMENSARBETE

Författare: Christoffer Smeds  
Utbildningsprogram och ort: Informationsteknik, Vasa  
Handledare: Mats Braskén

Titel: *Webbaserad bokningskalender och datorstyrning av värmesystem*

---

Datum 26.4.2011

Sidantal 31

Bilagor 1

---

### Abstrakt

Examensarbetet utfördes på begäran av Gerby Ungdomsförening r.f. som ville ha en bokningskalender för att enkelt hålla koll på när lokalen är uthyrd. Dessutom ville man ha styrning av värmesystemet kopplat till bokningskalendern. Arbetet bestod av att planera och utveckla datorprogram och webbsidor som sköter dessa uppgifter. En webbaserad bokningskalender skapades i PHP och den använder en MySQL-databas för att spara informationen. Programmet som styr uppvärmningen av lokalen skapades även det i PHP och kommunicerar med det elektroniska styrsystemet med hjälp av 1-Wire teknik.

---

Språk: svenska    Nyckelord: värmesystem, bokningskalender, 1-Wire, PHP

---

Förvaras: Theseus.fi och Tritonia, Vasa vetenskapliga bibliotek

## **BACHELOR'S THESIS**

Author: Christoffer Smeds  
Degree Programme: Information Technology  
Supervisor: Mats Braskén

Title: *Web-based booking calendar and computer control of heating system*

---

Date 26.4.2011                      Number of pages 31                      Appendices 1

---

### **Summary**

The thesis was done at the request of Gerby Ungdomsförening r.f. who wanted to have a booking calendar to easily keep track of when the property is leased. In addition, they wanted computer control of the heating system linked to the booking calendar. The work consisted of planning and developing computer programs and a website that takes care of these tasks. A web-based booking calendar was created in PHP and it uses a MySQL database to store data. The program that controls the heating was also created in PHP and it communicates with the electronic control system using 1-Wire technology.

---

Language: Swedish                      Key words: heating system, booking calendar, 1-Wire, PHP

---

Filed at: Theseus.fi and The Tritonia Academic Library, Vaasa

## OPINNÄYTETYÖ

Tekijä: Christoffer Smeds  
Koulutusohjelma ja paikkakunta: Tietotekniikka, Vaasa  
Ohjaaja: Mats Braskén

Nimike: *Web-pohjainen varauskalenteri ja lämmitysjärjestelmän ohjaus tietokoneella*

---

Päivämäärä 26.4.2011

Sivumäärä 31

Liitteet 1

---

### Tiivistelmä

Opinnäytetyö tehtiin Gerby Ungdomsförening r.f.:n pyynnöstä. He halusivat varauskalenterin, jonka avulla voi helposti pitää kirjaa varauksista. Sen lisäksi haluttiin liittää lämmitysjärjestelmän ohjaus varauskalenteriin. Työ perustui sen mahdollistavan tietokoneohjelman ja websivujen suunnitteluun ja kehittämiseen. PHP:llä luotiin webpohjainen varauskalenteri, joka käyttää MySQL-tietokantaa tiedon tallentamiseen. Myös lämmitystä ohjaava ohjelma luotiin PHP:llä. Ohjelma kommunikoi elektronisen ohjausjärjestelmän kanssa 1-Wire tekniikan avulla.

---

Kieli: ruotsi Avainsanat: lämmitysjärjestelmä, varauskalenteri, 1-Wire, PHP

---

Arkistoidaan: Theseus.fi ja Tritonia, Vaasan tiedekirjasto

# INNEHÅLLSFÖRTECKNING

ABSTRAKT

SUMMARY

TIIVISTELMÄ

1	INLEDNING .....	1
1.1	Gerby Ungdomsförening r.f. ....	1
1.2	Bakgrund.....	2
1.3	Uppgift.....	3
2	DET ELEKTRONISKA STYRSYSTEMET .....	5
3	SPRÅK OCH PROTOKOLL .....	7
3.1	Webb- och skriptspråk.....	7
3.1.1	HTML.....	7
3.1.2	CSS.....	8
3.1.3	PHP.....	8
3.2	SQL.....	10
3.3	1-Wire protokollet .....	10
4	TEKNIKER .....	12
4.1	MySQL .....	12
4.2	OpenWrt .....	14
4.3	1-Wire komponenter.....	15
4.4	OWFS .....	16
4.4.1	Owserver .....	17
4.4.2	Owfs .....	17
4.4.3	Owhttpd.....	18
4.4.4	Exempel på ett 1-Wire system .....	18
4.5	Lösenordshantering.....	18

5	UTFÖRANDE .....	20
5.1	Systemets uppbyggnad .....	20
5.1.1	Databasen .....	21
5.1.2	Bokningskalendern.....	22
5.1.3	Justerprogrammet.....	25
5.2	Användargränssnitt .....	26
6	RESULTAT OCH DISKUSSION .....	28
6.1	Resultat .....	28
6.2	Fortsatt utveckling .....	28
6.3	Reflektioner .....	29
7	KÄLLFÖRTECKNING .....	30

# 1 INLEDNING

Detta examensarbete utfördes på uppdrag av Gerby Ungdomsförening r.f.. Kontaktperson vid föreningen var Lucas Nyberg och handledare var Mats Braskén vid Yrkeshögskolan Novia.

## 1.1 Gerby Ungdomsförening r.f.

Gerby Ungdomsförening r.f. är en ideell förening som har som huvuduppgift att ordna aktiviteter åt barn och unga i Gerby- och Västervik området. Föreningen grundades 1928 och har sedan dess haft en aktiv del i vad som sker på området. Man har sedan slutet av 1920-talet haft en 440 m<sup>2</sup> stor lokal i centrala Gerby, se figur 1, vilken har verkat som centralpunkt för föreningens verksamhet. /1/



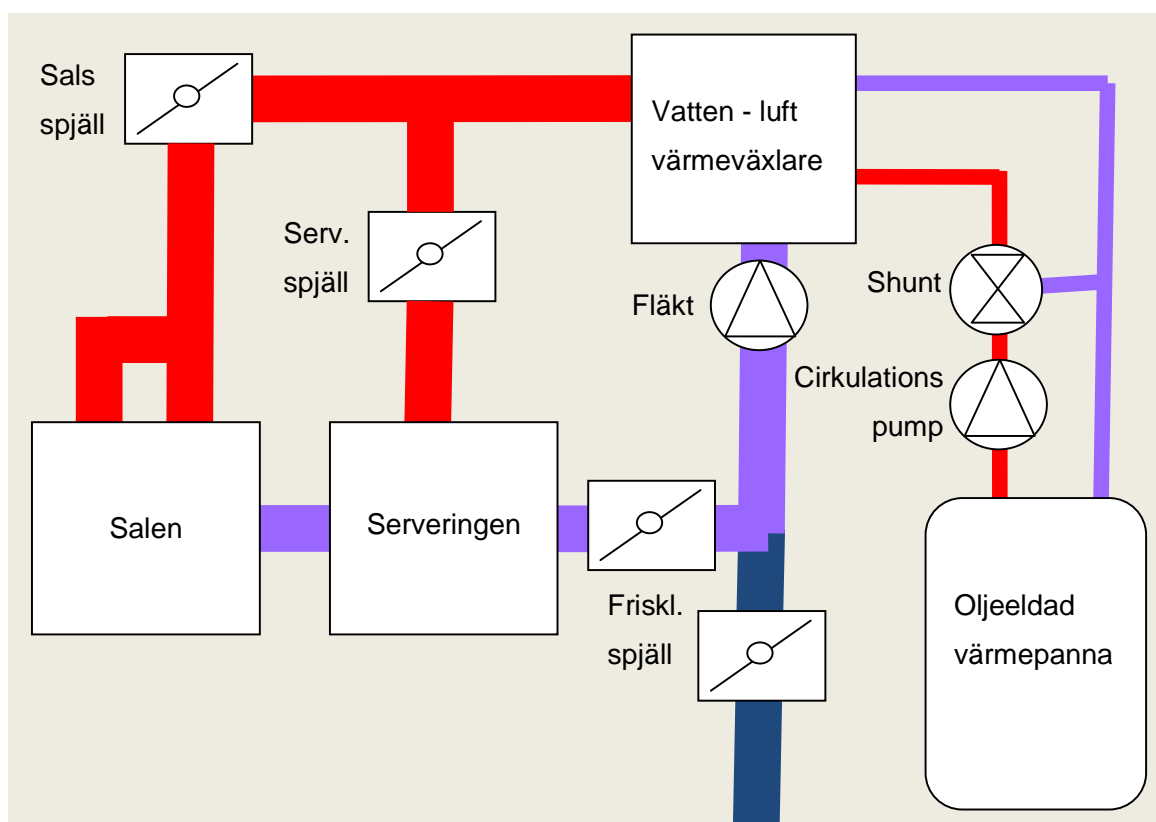
*Figur 1. Gerby Ungdomsförenings lokal, Lillbyvägen 28. /14/*

På senare år har intresset för ungdomsverksamhet svalnat och föreningen har allt svårare att hitta nya aktiva medlemmar som är beredda att ta över efter en åldrande styrelse. I dagens läge har föreningen ytterst lite egen verksamhet och det enda som håller fastigheten från att förfalla är det stora intresset hos allmänheten att hyra lokalen för diverse fester och sammankomster.

## 1.2 Bakgrund

Allteftersom det har blivit svårare att hitta nya medlemmar har de gamla medlemmarna fått fler och fler uppgifter att ta hand om. Till en av de mest krävande uppgifterna hör att sköta om så att lokalen är uppvärmd när det skall ordnas något där. Då lokalen inte är i användning håller man endast grundvärme i den för att bl.a. förhindra att vattenrören fryser och då lokalen skall användas värms den upp igen.

Värmesystemet består av en oljepanna, en vatten-till-luft värmeväxlare och ett system av luftkanaler som fördelar värmen mellan serveringsutrymmet och salen, se figur 2.



Figur 2. Värmesystemet.

Tidigare styrdes uppvärmningen med en timer, i vilken man kunde programmera in en veckas uppvärmningar åt gången. Systemet kunde då inte



själv ställa om så att t.ex. salen och serveringen hade olika temperaturer, utan detta måste ställas in manuellt inför varje bokning. Dessutom började den gamla tekniken ställa till med problem och allt oftare hände det sig att lokalen inte var uppvärmd då hyresgästerna kom dit.

Ett nytt elektroniskt styrsystem planerades under vintern 2006-2007 av Richard Granholm, som ett examensarbete vid Svenska Yrkeshögskolan. Det nya systemet ger fler möjligheter att kontrollera uppvärmningen genom att man kan styra systemet, endera via en kontrollpanel i pannrummet, eller med hjälp av en dator.

En annan tidskrävande uppgift inom föreningen är att sköta om uthyrningen av lokalen och fakturering av hyresgäster. Det saknades ett ordentligt system för att hålla reda på när lokalen var ledig och när den var uthyrd. Dessutom måste den person som skötte om uthyrningarna ständigt vara i kontakt med den som skötte uppvärmningen av lokalen då det skedde ändringar i bokningarna.

Det var arbetet med de här två uppgifterna som man hoppades underlätta genom att ha ett datorstyrt uppvärmningssystem, som får information direkt från bokningskalendern.

### **1.3 Uppgift**

Uppgiften kunde delas in i två deluppgifter; första deluppgiften var att skapa en webbaserad bokningskalender som är enkel och lättanvänd, men ändå innehåller alla uppgifter som behövs för att processera en bokning. Den som fakturerar skall inte behöva vara samma person som sköter om uthyrningen av lokalen, utan all nödvändig information skall gå att hämta ur kalendern. En enkel sammanfattning av när lokalen är bokad kommer att läggas upp på föreningens hemsida så att medlemmar och övriga intresserade kan se när lokalen är ledig.

I bokningskalendern skall man förutom tidsperioden för uthyrningen även kunna spara information om:

- vilken typ av evenemang som ordnas
- vem hyresgästen är
- vem som hyrt ut lokalen
- vilken temperatur som önskas

Andra deluppgiften var att skapa den programvara som skall kommunicera med det elektroniska styrsystemet som styr uppvärmningen av lokalen. Justerprogrammet skall hämta uppgifter från bokningskalendern om när lokalen skall värmas upp, vilken del av lokalen som skall värmas och vilken temperatur som skall hållas. Det räknar sedan ut hur värmesystemet skall ställas in för att på bästa sätt uppnå och sedan hålla de önskade temperaturerna. Programmet skickar sedan instruktioner, som säger vilka positioner olika spjäll och fläktar skall ha, åt det elektroniska styrsystemet.

Justerprogrammet skall kunna:

- kommunicera med kalenderservern på ett säkert sätt
- kommunicera med det elektroniska styrsystemet
- beräkna hur värmesystemet skall ställas in
- kontrollera temperaturerna i lokalen
- meddela någon person vid felsituationer

Examensarbetet är upplagt så att jag börjar med att beskriva det elektroniska styrsystemet. Sedan beskriver jag de språk, protokoll och tekniker som jag kommit i kontakt med vid utförandet av detta examensarbete. Efter det en kort beskrivning av lösningen till uppgifterna. Som avslutning reflekterar jag över hur arbetet med examensarbetet har förlöpt och hur man eventuellt kunde fortsätta utvecklingen av programmen.

## 2 DET ELEKTRONISKA STYRSYSTEMET

Det elektroniska styrsystemet är planerat så att det skall styras av en dator, men systemet innehåller även en analog reglerprocess så att det skall kunna fungera utan datorstyrning. Elektroniken är planerad som instickskort på ett backplane, likt i en dator där t.ex. ljudkort och grafikkort inte är integrerade i moderkortet utan sticks in i portar på moderkortet. Detta underlättar vid felsökning och underhåll, eftersom korten kan bytas sinsemellan samt felsökas enskilt. /3/

Styrsystemet har fem utgångar för att styra de olika komponenterna i värmesystemet, se figur 2. Sammanfattningsvis är komponenterna: salsspjället, serveringsspjället, friskluftsspjället, shunten och fläkten.

Styrsystemet har sju ingångar för att ta emot värden för:

- salsspjällets position
- serveringsspjällets position
- friskluftsspjällets position
- shuntens position
- fläktens läge
- önskade temperaturer
- vilket läge systemet skall vara i

Ingångarna är 1-Wire kompatibla komponenter som lätt kan styras av en dator.

Styrsystemet har åtta olika lägen. Beroende på vilket läge systemet är i, bestäms värdena på utgångarna på olika sätt. Läge 1, 2 och 3 är de lägen som styrs mer eller mindre av datorn, nedan förklaras dessa närmare, i bilaga 1 finns samtliga lägen förklarade.

### Läge 1

I det här läget styr datorn allting. Temperaturerna i lokalen mäts med hjälp av digitala givare och justerprogrammet ställer in fläktar och spjäll för att uppnå

önskade temperaturer givna i bokningskalendern på ett snabbt och ekonomiskt sätt. Detta läge kommer att vara det läge som används för det mesta.

### **Läge 2**

Samma som läge 1 men i stället för att ta önskade temperaturer från bokningskalendern, så tar datorn temperaturerna från en potentiometer på kontrollpanelen. Detta läge används då man inte gjort någon bokning, utan då man endast vill ändra temperaturen tillfälligt.

### **Läge 3**

Systemet får endast önskade temperaturer från datorn. Temperaturerna i lokalen mäts med hjälp av analoga givare och styrsystemet ställer om fläktar och spjäll för att uppnå önskade temperaturer. I detta läge är regleringen relativt grov och kan nästan anses fungera som ON/OFF-lägen. Datorn kan från detta läge ställa om till läge 1 eller 2.

Systemet har även en säkerhetsmekanism som känner av om kontakten till datorn bryts och sätter då systemet i ett felsäkert läge som håller en förinställd temperatur i lokalen.

Det elektroniska styrsystemet är ännu inte fullständigt färdigt och detta innebär tyvärr att provkörningen av justerprogrammet, vilket är en del av detta examensarbete, kommer att skjutas fram.

## 3 SPRÅK OCH PROTOKOLL

### 3.1 Webb- och skriptspråk

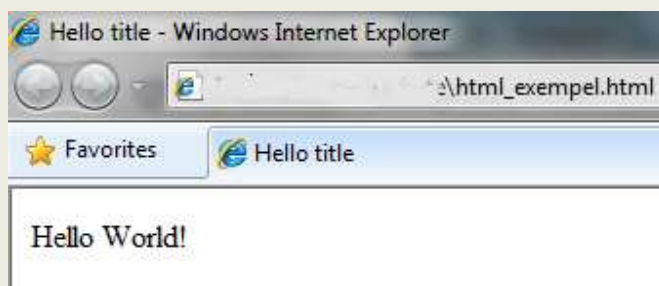
I detta kapitel förklaras de grundläggande språken som användes vid genomförandet av examensarbetet. Språken är HTML, CSS och PHP, dessa är några av de viktigaste byggstenarna man använder sig av då man skapar webbsidor.

#### 3.1.1 HTML

HTML är en akronym av *Hyper Text Markup Language* och är ett märkspråk och en standard för hur man strukturerar t.ex. text och bilder på en webbsida. Ett märkspråk används för att berätta för ett program hur informationen skall visas upp för användaren. I HTML använder man vinkelparenteser för att avskilja styrkoderna från den egentliga informationen. Den nyaste versionen av HTML-standarderna är HTML 5, den är fortfarande i utvecklingsstadiet och därför kommer detta examensarbete att basera sig på HTML 4.01. Att skapa en enkel webbsida med HTML går snabbt och enkelt, se kodexempel 1. /7/

*Kodexempel 1. Koden för en enkel HTML sida och resultatet.*

```
<html>
<head>
  <title>Hello title</title>
</head>
<body>
  <p>Hello World!</p>
</body>
</html>
```



### 3.1.2 CSS

CSS är en akronym av *Cascading Style Sheets* och är ett språk som man använder för att beskriva presentationsstilen för ett dokument. Genom att använda CSS-filer vill man åtskilja information om hur innehållet skall presenteras från själva innehållet. Dessutom minskar man mängden kod som varje HTML-sida behöver innehålla genom att hänvisa till en CSS-fil istället för att ställa in värdena på varje sida enskilt. Man får enkelt ett snyggt helhetsutseende på sin webbplats. I en CSS-fil bestämmer man vanligtvis åtminstone typsnitt, textstorlek, färger, marginaler och bakgrundsbilder, se kodexempel 2. /16/

*Kodexempel 2. Två typiska CSS klasser.*

```
body {  
  background-color: lightgrey;  
}  
  
p {  
  font-family: Arial;  
  font-size: 14px;  
  color: black  
}
```

Hello World!

### 3.1.3 PHP

PHP är en rekursiv akronym för *PHP: Hypertext Preprocessor*. Det är ett skriptspråk som används främst på webbservrar för att skapa webbsidor med dynamiskt innehåll, t.ex. nyhetssidor. Dynamiskt innehåll betyder att innehållet hämtas exempelvis från en databas och skriptet skapar en webbsida kring innehållet. I och med att skriptet hämtar aktuell information varje gång en användare begär sidan, så kan man enkelt byta ut informationen som skall visas på sidan utan att behöva gå in och koda om webbsidan. PHP har stöd för många olika databashanterare, t.ex. Oracle, Microsoft SQL Server och MySQL.

/11/

Ett alternativ till PHP skulle vara *Active Server Pages* (ASP) som är ett konkurrerande språk. Huvudorsakerna till varför jag valde PHP över ASP för detta examensarbete var två:

1. Servern som föreningens existerande hemsidor ligger på har redan stöd för PHP. ASP-stöd skulle ha kostat extra och eftersom PHP är lika effektivt för detta ändamål ansågs detta onödigt.
2. Det finns en kommandotolkapplikation som kör PHP-skript direkt utan att man behöver ha en webbrowser installerad. Detta öppnade för möjligheten att även skriva justerprogrammet i PHP som är resurssnålt och går bra att köra på en enkel dator med begränsade resurser.

PHP-koden är likadan oavsett om man kör den på en webbrowser eller direkt i kommandotolken. T.ex. `<?php echo 'Hello World!'; ?>` skriver ut "Hello World!" på en webbsida eller i kommandotolken.

*Kodexempel 3. Koden från exempel 1 där titeln och innehållet ersatts med PHP-variabler, slutresultatet är det samma.*

```
<?php
$page_title = 'Hello title';
$content = 'Hello World!';
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" http://www.w3.org/TR/html4/strict.dtd>
<html>
  <head>
    <title><?php echo $page_title; ?></title>
  </head>
  <body>
    <p><?php echo $content; ?></p>
  </body>
</html>
```



## 3.2 SQL

SQL är en akronym för *Structured Query Language* och är ett standardiserat språk för att t.ex. hämta, lägga till, ändra på eller ta bort data ur en tabell i en relationsdatabas. Förutom att ändra på data i tabeller kan man även skapa tabeller, ta bort tabeller, ändra på tabellernas definition o.s.v. med hjälp av SQL-kommandon. /13/

*Kodexempel 4. Några enkla SQL-kommandon för att lägga till, hämta och ta bort data ur en relationsdatabas.*

```
INSERT INTO BankKunder (namn, kontoNr) VALUES ('Kalle', '111-22222');  
SELECT namn, kontoNr FROM BankKunder WHERE namn = 'Kalle';  
DELETE FROM BankKunder WHERE id = 1;
```

Kommunikation med MySQL-databasen som används i detta examensarbete bygger på SQL-kommandon.

## 3.3 1-Wire protokollet

De språk som hittills förklarats har varit ADB-specifika språk, nu då vi kommer till 1-Wire protokollet går vi över på mera elektroniskspecifika protokoll. 1-Wire kommer huvudsakligen att användas vid kommunikationen mellan datorn och det elektroniska styrsystemet. Det kommer även att användas då datorn mäter temperaturen i lokalen med hjälp av digitala temperatursensorer.

1-Wire är ett kommunikationsprotokoll för kommunikation mellan små och enkla komponenter som t.ex. digitala temperaturmätare. Det erbjuder en låghastighets databuss som förutom att överföra data, även förser komponenterna med ström via en enda tråd, därav namnet.

Bussen använder egentligen två trådar, en för dataöverföring/ström och en för jord. För att komponenterna inte skall bli utan ström då tråden används för överföring av data har varje komponent i 1-Wire systemet en inbyggd



kondensator. En 1-Wire komponent kan endera placeras självständigt eller på ett kretskort som en del av ett komplett system.

Några av fördelarna med ett 1-Wire system är att komponenterna är billiga, det är enkelt att ansluta nya komponenter och systemet är självkonfigurerande, vilket betyder att det automatiskt upptäcker nya komponenter som kopplats in och ställer in dessa. Nackdelarna är att det är relativt långsamt och man behöver en huvudenhet som t.ex. OW Server, att styra systemet med. /9/

## 4 TEKNIKER

Här förklaras de tekniker som använts vid genomförandet av uppgifterna samt förklaringar varför man har valt just dessa tekniker.

### 4.1 MySQL

En MySQL-databas kommer att användas som grund för bokningskalendern. Orsakerna till att jag valde denna databashanterare var att föreningens befintliga hemsidor finns på en server som har stöd för MySQL, samt att PHP har ett inbyggt bibliotek för att enkelt kunna kommunicera med en MySQL-databas.

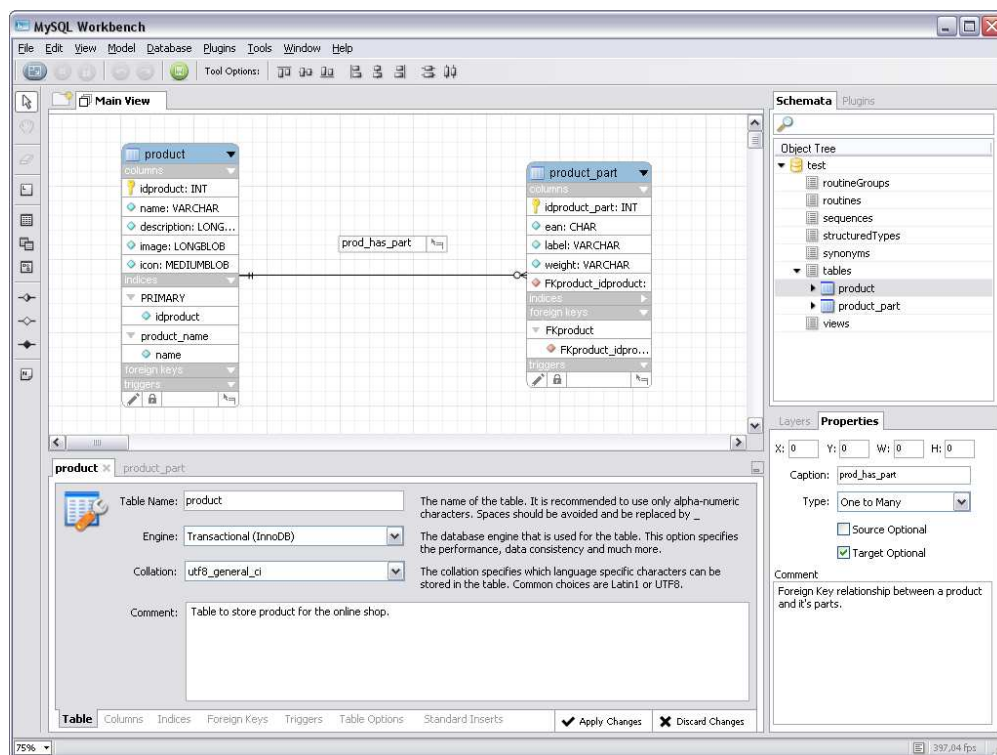
MySQL är en databashanterare som är baserad på öppen källkod och är enligt egen utsago världens mest populära sådana. Första versionen av MySQL skapades i mitten av 1990-talet av det svenska bolaget MySQL AB. Projektet såldes senare till Sun Microsystems, som köptes upp av Oracle som äger projektet idag. Gratisprogram som behöver en databas att arbeta mot, använder ofta MySQL just för att standardversionen är gratis. Det finns även flera olika betalversioner av databashanteraren för kommersiellt bruk. Betalversioner används bl.a. av Nokia, Wikipedia, Google och Yahoo. /5/

Databashanteraren går att köra på alla stora operativsystem, t.ex. Windows, Linux och en mängd olika Unix operativsystem. Det finns MySQL-bibliotek i de flesta stora programmeringsspråken för att man enkelt skall kunna koda program som kör mot en databas. Även PHP som kommer att användas i detta examensarbete har ett MySQL-bibliotek.

Databashanteraren följer till stor del SQL-standarden och har stöd för bl.a. foreign keys för att koppla ihop data i olika tabeller; joins för att sammanfoga tabeller; triggers för att automatisera händelser i databasen och stored procedures för att spara funktioner som ofta körs mot databasen i databasen för lätt åtkomst. Den har även stöd för de vanligaste datatyperna: integer, numeric, boolean, char, varchar, date och timestamp. MySQL har inbyggt gränssnitt för

programmering av t.ex. stored procedures i en mängd olika programmeringsspråk, däribland C/C++, Java och .Net. /6/

MySQL har även utvecklat ett eget program för att underlätta vid planeringen av databaser kallat *MySQL Workbench*, se figur 3.



Figur 3. *MySQL Workbench*.

Utvecklingen med att effektivisera databashanteraren pågår hela tiden. Den senaste versionen, MySQL 5.5, är snabbare än tidigare versioner tack vare sitt stöd för att använda alla processorer i dagens datorsystem som ofta har fler än en processor, s.k. multi-processing hårdvara. Enligt MySQL själva har prestanda ökat mångdubbelt i den nya versionen. /7/

Av de egenskaper som skiljer MySQL från övriga databashanterare, kan t.ex. nämnas multipla lagringsmotorer, vilket ger databasadministratören möjlighet att själv välja vilken motor som passar olika tabeller bäst. /6/

## 4.2 OpenWrt

Justerprogrammet som styr värmesystemet kommer att köras på en router som har konfigurerats med en OpenWrt-firmware.

En router är en nätverksenhet som sammankopplar datorer i ett nätverk och styr datatrafiken mellan datorerna med hjälp av en routingtabell. I routingtabellen sparas information om vilken väg som är snabbast för att överföra data till en viss dator. En enkel router har ofta idag även inbyggt t.ex. ett ADSL-modem, en brandvägg och en DHCP-server. /12/

OpenWrt är en liten Linux-distribution som är populär att installera på routrar för att få större frihet att styra över ovannämnda funktioner och även utöka med nya funktioner. I distributionen ingår förutom Linux kerneln en samling av diverse programpaket. I Linux-system använder man olika pakethanterare som sköter om installationen av program, OpenWrt har en egen pakethanterare kallad opkg. OpenWrt används oftast genom kommandolinje gränssnittet sh, men man kan även installera bash eller ett grafiskt användargränssnitt, t.ex. LuCI, se figur 4, eller X-Wrt om man vill. /8/



Figur 4. OpenWrts webbkonfigureringsverktyg

OpenWrt-projektet blev till efter att Linksys skapat sin firmware till deras trådlösa router WRT54G under GPL-licensen, vilket betydde att de måste

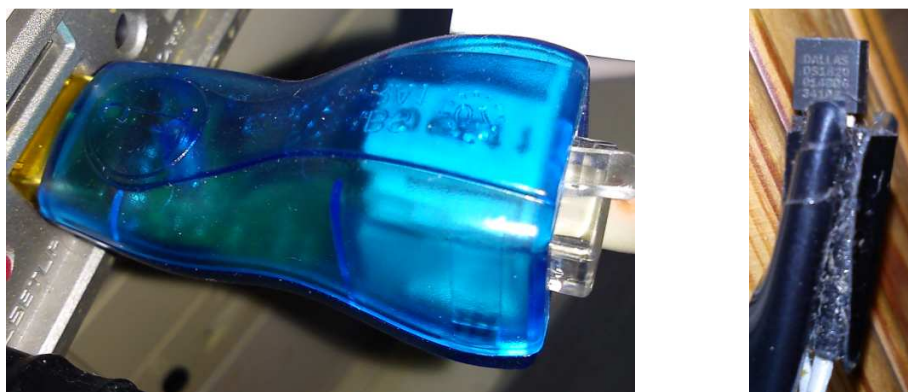
släppa källkoden till firmwaren fri. Detta gjorde att systemutvecklare kunde skapa sina egna versioner som innehöll funktioner och program som inte fanns i de normala routrarna som man kunde köpa i butiken. Först stödde OpenWrt endast Linksys WRT54G router-serien, men senare har stödet utvidgats till andra kretsuppsättningar och tillverkare.

Förutom att erbjuda de funktioner som standardrouter firmwaren har, ger OpenWrt även tillgång till bl.a. routingtabellerna, brandväggar, port forwarding, ip tunneling och tusentals andra program skapade för Linux. Den intressantaste egenskapen hos OpenWrt är möjligheten att skriva helt egna program och skript. Programmen och skripten kan sedan köras som på vilken vanlig dator som helst.

Många routrar har en USB-port som är tänkt att ge användaren möjligheten att konfigurera nätverksinställningar utan att behöva ta kontakt via nätverksporten. Med OpenWrt installerat på routern kan man t.ex. dela ut en USB-skrivare till nätverket eller koppla in en 1-Wire bus masterenhet med hjälp av USB-porten.

### 4.3 1-Wire komponenter

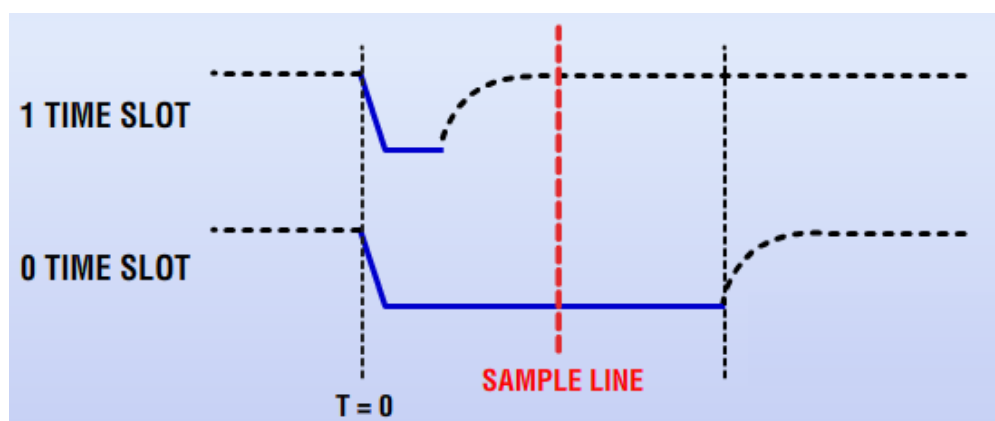
Komponenterna i ett 1-Wire system delas upp i två kategorier: master och slavar. Master är den som initierar kommunikationen i systemet, och det kan vara t.ex. en bus-master som styrs av en dator via USB- eller serieporten, se figur 5a.



Figur 5a, b. En 1-Wire USB-adapter och en DS18S20 temperatursensorer.

Övriga komponenter i systemet är s.k. slavar. För att skilja åt alla slavar har varje komponent ett 8 byte långt id-nummer. Den första byten identifierar vilken typ av komponent det är, de sex påföljande byten är en unik id och den sista byten är en kontroll-byte. Alla slavar kopplas in parallellt på en och samma kabel. Olika mätkomponenter i ett 1-Wire system kan vara t.ex. temperatur-, luftfuktighets-, spännings-, strömsensorer och brytare, se figur 5b. Det finns även mer avancerade komponenter som t.ex. stöder kryptering av data och komponenter som förser stora 1-Wire-system med ström då inte masterns strömförsörjning räcker till. Vanligen använder man en CAT5-kabel för att koppla ihop alla komponenterna. /1, 10/

Mastern och slavar kommunicerar med varandra genom att dra ner signalen till jord en viss tid beroende på om man vill skicka en 1:a eller 0:a, se figur 6. /4/



Figur 6. 1-Wire komponenter skickar en 1:a eller en 0:a. /2/

1-Wire bussen har en teoretisk maximalhastighet på 16,3 kbit/s. /1/

## 4.4 OWFS

OWFS, *1-Wire File System*, är ett programpaket som gör det enkelt att administrera och använda ett 1-Wire system från en Linux-dator. /10/

#### 4.4.1 Owserver

Owserver är den första och viktigaste delen i OWFS-paketet, den är backend-komponenten. Den har en öppen tcp/ip-port som ger klientprocesserna tillgång till bussen. Klientprocesserna kan vara endera lokalt på samma dator, eller så kan de ta kontakt över ett nätverk, t.ex. Internet. Owserver har stöd för säker överföring av data t.ex. genom en SSH-tunnel.

Owserver kommunicerar med buss-mastern via en USB- eller serieport. Klienterna kan även ta direkt kontakt till buss-mastern, men åtkomsten till USB- eller serieporten är begränsad av operativsystemet till en åt gången. Då är det bättre att använda Owserver då den tillåter flera klientanslutningar på samma gång.

Owserver frågar med jämna mellanrum av komponenterna vilka deras värden är och sparar dessa i ett cacheminne för att snabba på åtkomsttiden för klientprogrammen. Hur ofta är ställbart och är vanligen mellan 1 och 10 sekunder. Klienterna kan också be Owserver om det aktuella värdet, i vilket fall Owserver frågar komponenten om värdet och klienten får vänta en stund.

#### 4.4.2 Owfs

Owfs är den del av programpaketet som ger möjligheten att skapa ett virtuellt filsystem av komponenterna med hjälp av fuse, *Filesystem in Userspace*, som finns inbyggd i Linux-kerneln. I praktiken skapar owfs-programmet ett virtuellt filsystem där komponentens unika id är direktoratnamnet och komponentens egenskaper visas som filer i direktoratet.

Hela 1-Wire systemet monteras på en särskild plats i filsystemet och användarna kan komma åt komponenterna genom att använda vanliga filkommandon, t.ex. *cat* för att läsa innehållet i en fil. Alla filer och direktorat är endast virtuella och man kan inte skapa, ta bort eller döpa om filerna.

### 4.4.3 Owhttpd

Owhttpd är en liten lokal webbserver som upprätthåller en webbplats där man hittar data om och från komponenterna i ett 1-Wire system kopplat till en Owserver. Strukturen på webbsidan som visas är densamma som för filsystemet, först visas alla komponenter i systemet sedan kan man gå in i varje komponent och se dess olika värden.

### 4.4.4 Exempel på ett 1-Wire system

Enklast möjliga 1-Wire system skulle vara att ta t.ex. en DS18S20 temperatursensor, koppla ihop den med hjälp av en CAT5-kabel med en 1-Wire USB-adapter och ansluta USB-adaptern till en Linux-dator på vilken man installerat OWFS-programpaketet. Systemet är som tidigare nämnts självkonfigurerande så inga inställningar behöver göras. Nu kan man med enkla Linux-kommandon läsa av temperatursensorn eller så kan man med en webbläsare surfa in på webbservern och kontrollera temperaturen därifrån.

## 4.5 Lösenordshantering

I det här examensarbetet kommer lösenorden att sparas som en MD5 hash-summa i databasen. MD5 är en envägs-kryptering och dess hash-funktion kan mycket förenklat förklaras med ett exempel. Man tar lösenordssträngen, t.ex. "1234" och adderar ihop talen och får hash-summan "10". Summan sparas i databasen och när en användare försöker logga in räknas hash-summan för det givna lösenordet ihop igen och jämförs med den i databasen sparade summan. Envägs betyder att man kan endast kryptera och inte dekryptera lösenordet, man kan t.ex. inte från "10" bestämma vilka tal som användes för att få summan.

Dock finns det ordlistor som innehåller hash-summorna för en stor mängd ord, upp till miljontals ord. Dessa listor kan användas för att jämföra summorna och



på det viset hitta rätt lösenord. Därför är det viktigt att inte använda vanliga ord och teckenföljder som lösenord.

För att ännu öka på säkerheten så kan man använda sig av en metod som kallas att "salta lösenordet". Det innebär att man lägger till en textsträng med på förhand slumpmässigt valda tecken före och efter det av användaren valda lösenordet. Det minskar risken för att s.k. ordliste-attacker lyckas. /15/

Orsaken varför MD5 valdes var att både PHP och MySQL har inbyggda funktioner för att räkna ut hash-summan för just MD5-kryptering.

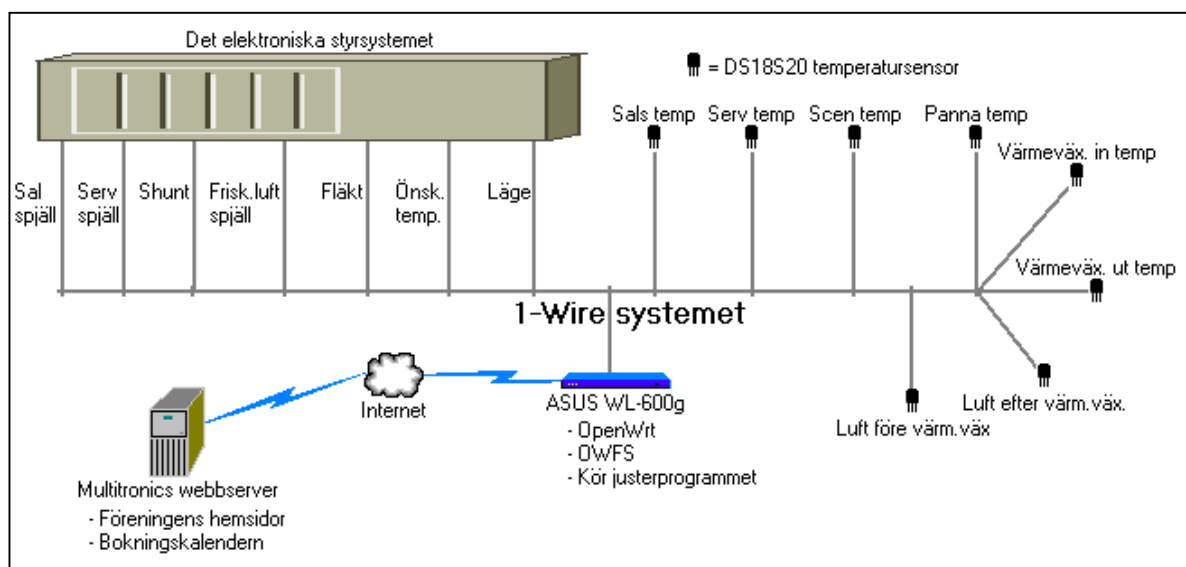
## 5 UTFÖRANDE

### 5.1 Systemets uppbyggnad

Föreningen har från tidigare hemsidor på Multitronics webbserver, de är skrivna i vanlig HTML-kod och utseendet styrs med hjälp av CSS. Bokningskalendern kommer att placeras på samma server som förutom PHP-stöd även har stöd för att köra en MySQL-databas. Tills åtkomst till Multitronics server blir möjlig, används Novias Loke-server som utvecklingsmiljö.

Justerprogrammet kommer att köras på en OpenWrt-distribution som konfigurerats med bl.a. OWFS programpaketet. Hårdvaran är en ASUS WL-600g router som finns i föreningslokalen, se figur 7.

Runtom i lokalen finns temperatursensorer av typen DS18S20 som kopplats in på 1-Wire systemet, se figur 7. Dessa används av justerprogrammet för att mäta temperaturen i lokalen.



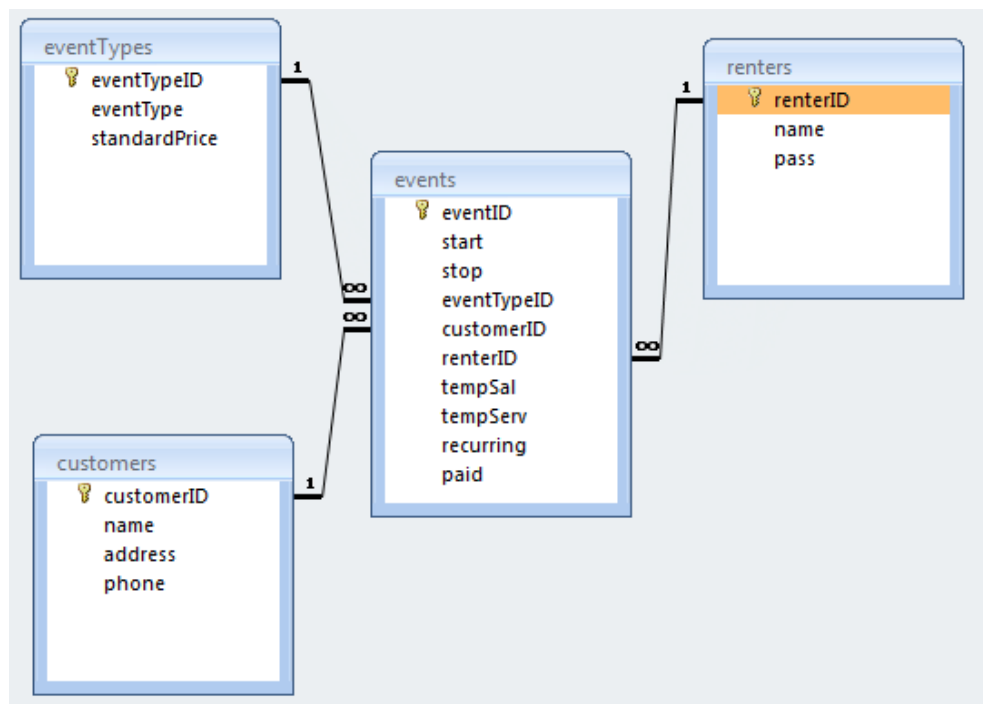
Figur 7. Överblick över styrsystemet.

### 5.1.1 Databasen

Databasen som ligger som grund för bokningskalendern är en MySQL-relationsdatabas som består av fyra tabeller, se databasschemat i figur 8. Databasen har designats utgående från specifikationerna i kraven för bokningskalendern, med vissa tillägg för t.ex. lösenordshanteringen för inloggning till webbsidorna.

För att komma in i bokningskalendern, måste användaren vara registrerad som en uthyrare i tabellen *renters*. Det krypterade lösenordet, som användaren själv kan välja, sparas här som en 32 tecken lång teckensträng. Varje tabell har en id-kolumn som är en automatisk numreringskolumn, undantaget är *renters* tabellen där id kommer att användas vid inloggning och därför valdes personens initialer som id i stället.

En bokning kan vara återkommande, vilket betyder att den inte behöver faktureras efter varje gång utan man fakturerar t.ex. per termin i stället. En kolumn för att markera vilka bokningar som redan betalats inkluderades också för att minska risken för att redan betalda bokningar faktureras.



Figur 8. Databasschema.

Databasen förbereddes med ett antal uthyrare och några vanliga kunder. Olika evenemangstyper matades in samt standardpriset för dessa. Vartefter att man behöver komplettera med nya uthyrare, kunder och evenemangstyper kan man göra detta via bokningskalendern.

### 5.1.2 Bokningskalendern

Koden för bokningskalendern är uppdelad i fem PHP-filer. Funktionerna för att ansluta till databasen och hantera data till och från databasen finns i *database.php*, som refereras till i de övriga filerna. Kodexempel 5 visar en funktion som byggs in i databasklassen för att ansluta till databasen. Funktionen returnerar ”true” om anslutningen lyckades.

*Kodexempel 5. Ansluter till databasen på den lokala datorn.*

```
private function connectDb()
{
    $p = mysql_connect('localhost', $this->user, $this->password);
    if (!$p)
        return false;
    else
    {
        $d = mysql_select_db($this->database);
        if (!$d)
            return false;
    }
    return true;
}
```

Varje gång ett kommando skall utföras på databasen öppnas anslutningen, kommandot körs, data returneras och sedan stängs anslutningen automatiskt.

Behandling av bokningar sker på *event.php* sidan. Sidan har tre indelningar: skapa ny bokning, visa befintlig bokning och spara bokning. Med hjälp av de argumenten som ges då sidan anropas, visas endera ett tomt formulär, ett färdigt ifyllt formulär, eller en bekräftelse av att bokningen sparats.

Borttagning av en bokning görs av *deleteEvent.php* som anropas med ett *bokningsid* som argument. I kodexempel 6 används funktionen *deleteEvent(\$id)*

för att ta bort en bokning ur databasen. Den inbyggda PHP-funktionen *mysql\_real\_escape\_string(string)* används på alla ställen där användaren kan ge in data, för att kontrollera att data som ges in inte innehåller något förbjudet tecken eller kod. Tecken som kan vara skadliga om de tillåts i en fråga som körs mot databasen är s.k. wildcards, t.ex. stjärna (\*). T.ex. om någon skulle anropa *deleteEvent.php* med *bokningsid = '\*'*, så skulle alla bokningar raderas.

*Kodexempel 6. PHP-koden i deleteEvent.php*

```
<?
require("database.php");
$db = new CDatabase('user', 'password', 'database');
$id = mysql_real_escape_string($_GET['ID']);
if ($db->deleteEvent($id))
    echo "Bokningen raderades!";
else
    echo "Bokningen kunde inte raderas!";
?>
```

Funktionen *deleteEvent(\$id)* är en del av databasklassen och dess implementering visas i kodexempel 7.

*Kodexempel 7. deleteEvent(\$id)'s implementering*

```
public function deleteEvent($ID)
{
    if ($this->connectDb())
    {
        $sql = "DELETE FROM events WHERE eventID = '$ID'";
        $res = mysql_query($sql);
        if (!$res)
            return false;
        else
            return true;
    }
    else
        return false;
}
```

Formulär för att lägga till uthyrare, kunder och evenemangstyper finns på *add.php*-sidan. Genom att kontrollera värdet på en invariabel bestäms vilket formulär som skall visas. I kodexempel 8 visas hur sidans titel beror på vad användaren har valt att lägga till.

*Kodexempel 8. Skapar HTML-sidan beroende på variabeln "item"*

```
if ($_GET['item'] == 'typ')
    echo "<title>Lägg till typ</title>";
elseif ($_GET['item'] == 'arrangör')
    echo "<title>Lägg till arrangör</title>";
elseif ($_GET['item'] == 'uthyrare')
    echo "<title>Lägg till uthyrare</title>";
```

Huvudsidan *index.php* binder ihop alla funktionerna i bokningskalendern genom att länka till de övriga sidorna. Sidan innehåller även en tabell över alla bokningar som är inmatade i kalendern.

Föreningens befintliga hemsidas CSS-fil har återanvänts för att ge ett enhetligt utseende åt sidorna.

På webbservern finns även ett PHP-skript som plockar ut bokningarna för de följande sju dagarna ur bokningskalendern och sparar dessa i en textfil, se kodexempel 9. Skriptet körs med hjälp av *crontab*, som är en fil inbyggd i Linux där användaren kan lägga till jobb som man vill köra regelbundet. Skriptet körs varje hel timme för att man skall få med sista minuten-ändringar.

*Kodexempel 9. Plockar ut bokningarna ur databasen och sparar i en textfil.*

```
$query = "SELECT eventID, start, stop, tempSal, tempServ"
        ." FROM events WHERE stop > now()";
$result = mysql_query($query);
$rows = mysql_num_rows($result);

if ($rows > 0)
{
    $file = fopen('events.txt', 'w');

    for ($row=0;$row<$rows;$row++)
    {
        $data = mysql_fetch_row($result);
        fwrite($file, trim($data[0]). "; "
            . trim($data[1]). "; "
            . trim($data[2]). "; "
            . trim($data[3]). "; "
            . trim($data[4]). ";\n");
    }
    fclose($file);
    mysql_free_result($result);
}
```

### 5.1.3 Justerprogrammet

Justerprogrammet består av en samling PHP-skript som körs med hjälp av *crontab* på OpenWrt-routern. Första delen av programmet hämtar textfilen med bokningarna från webbservern en gång i timmen, fem minuter efter att filen skapats på servern, genom en SSH-tunnel som skapas inför varje överföring och stängs sedan. Programmet skriver aktuell tid och datum i en loggfil på servern, så att man kan gå in och kontrollera att hämtningen skett.

Skriptet som skapar textfilen på servern kollar loggfilen för att se om justerprogrammet hämtat filen. Om den inte hämtats inom de senaste tre timmarna skickas ett e-postmeddelande till en på förhand vald adress för att meddela om felet.

Själva justerprogrammet börjar med att kontrollera vilket läge det elektroniska styrsystemet är i. Sedan bestämmer systemet vilka de önskade temperaturerna i salen samt i serveringen är, om systemet är i läge 1 eller 3 tar programmet temperaturerna från bokningskalenderns textfil, om det är i läge 2 tar programmet dem från potentiometern på kontrollpanelen. Då programmet kontrollera i textfilen om det finns någon bokning för den aktuella tiden och det inte hittar någon bokning, vilket är det vanligast fallet, så ställs de önskade temperaturerna till 4 °C i salen och 7 °C i serveringen, den s.k. grundvärmen.

Efter att programmet bestämt temperaturerna mäter programmet de verkliga temperaturerna i lokalen och beräknar vilka positioner spjäll och fläktar skall ha för att på bästa sätt uppnå och behålla de önskade temperaturerna. Uträkningar för att hur man skall ställa spjäll och fläktar effektivast fanns till stor del från tidigare och har tagits fram genom testning över en lång tidsperiod. Temperaturer och lägen för spjäll och fläktar fanns loggade för flera år bakåt i tiden och genom att analysera dessa kunde man avgöra när det lönar sig att börja dra ner på fläkthastighet, ställa upp eller ner shunten, ställa om spjäll o.s.v.

Justerprogrammet kommunicerar med det elektroniska styrsystemet genom ett 1-Wire system där varje in- och utgång i styrsystemet har en egen id. Programmet läser av och skriver in nya värden åt styrsystemet. Om styrsystemet använder dessa värden eller om det beräknar egna värden beror på i vilket läge systemet är. I läge 1 och 2 använder styrsystemet direkt de värden för spjäll, fläktar o.s.v. som det får från datorn. Men i läge 3 använder det endast de önskade temperaturerna och beräknar själv resten.

## **5.2 Användargränssnitt**

Systemet kommer först och främst att styras via bokningskalendern, justerprogrammet har därför inte fått något användargränssnitt. Om man vill styra uppvärmningen utan att gå via bokningskalendern måste man ställa in värmen via kontrollpanelen i pannrummet. Orsaken till att man valt att systemet skall fungera på detta sätt är helt enkelt att man vill minska på eventuella inställningsmisstag, d.v.s. att användaren har ändrat på någon parameter som gör att systemet inte fungerar som det skall nästa gång lokalen skall värmas upp.

Bokningskalenderns användargränssnitt har designats för att det skall vara så enkelt som möjligt att använda. Endast grundläggande funktioner som att se bokningarna, ändra på bokningarna och lägga till bokningar har implementerats.

Föreningens befintliga hemsidors mjuka och bekväma utseende återfinns i bokningskalendern och man får snabbt en överblick av bokningsläget.



### Bokningar av UF-lokalen

[Registrera en bokning](#)

ID:	Start:	Stop:	Typ:	Arrangör:	Uthyrd av:	Temp sal:	Temp serv:	Aterkom.:	Betalt:	
1	2011-03-01 18:00:00	2011-03-01 20:30:00	Öppet Hus	Gerby UF	Christoffer Smeds	20	21	1	0	<a href="#">Ta bort</a> <a href="#">Ändra på</a>
22	2011-03-03 19:44:00	2011-03-03 20:44:00	Fest	Gerby UF	Anette Mouts	20	21	0	0	<a href="#">Ta bort</a> <a href="#">Ändra på</a>
3	2011-03-05 12:00:00	2011-03-06 15:00:00	Fest	Matti Meikäläinen	Christoffer Smeds	18	21	0	1	<a href="#">Ta bort</a> <a href="#">Ändra på</a>
2	2011-03-07 12:00:00	2011-03-07 16:00:00	Eftis	Settlement Vestis	Anette Mouts	20	21	1	0	<a href="#">Ta bort</a> <a href="#">Ändra på</a>
4	2011-03-08 18:00:00	2011-03-08 20:30:00	Öppet Hus	Gerby UF	Christoffer Smeds	20	21	1	0	<a href="#">Ta bort</a> <a href="#">Ändra på</a>
24	2011-03-10 18:00:00	2011-03-10 20:58:00	Öppet Hus	Gerby UF	Christoffer Smeds	20	21	0	0	<a href="#">Ta bort</a> <a href="#">Ändra på</a>
27	2011-03-10 19:47:00	2011-03-10 20:47:00	Bröllop	Gerby UF	Christoffer Smeds	20	22	0	0	<a href="#">Ta bort</a> <a href="#">Ändra på</a>
32	2011-03-10 20:41:00	2011-03-11 00:00:00	Bröllop	Gerby UF	Anette Mouts	20	21	0	1	<a href="#">Ta bort</a> <a href="#">Ändra på</a>
25	2011-03-11 18:00:00	2011-03-11 19:00:00	Ungdomsdans	Gerby UF	Christoffer Smeds	18	21	1	1	<a href="#">Ta bort</a> <a href="#">Ändra på</a>
26	2011-03-17 15:00:00	2011-03-17 21:00:00	Eftis	Settlement Vestis	Anette Mouts	18	21	0	0	<a href="#">Ta bort</a> <a href="#">Ändra på</a>
23	2011-04-03 19:47:00	2011-04-03 20:47:00	Öppet Hus	Gerby UF	Christoffer Smeds	20	21	0	0	<a href="#">Ta bort</a> <a href="#">Ändra på</a>

Figur 9. Bokningskalenderns första sida.

I figur 9 visas vad man ser då man surfar in till bokningskalendern, det är en enkel uppräknig av de kommande bokningar som är inmatade i databasen. Härifrån kan man via länken högst upp på sidan registrera en ny bokning, se figur 10a, eller så kan man med hjälp av länkarna till höger om bokningarna ändra på en redan registrerad bokning, se figur 10b.

#### Registrera bokning:

Boknings ID:

Start:

Stop:

Typ:  [Lägg till](#)

Arrangör:  [Lägg till](#)

Uthyrd av:  [Lägg till](#)

Temp sal:  (default: 20)

Temp serv:  (default: 21)

☐ Återkommande

☐ Betalt

[Gå tillbaka](#)

#### Ändra på bokning:

Boknings ID:

Start:

Stop:

Typ:  [Lägg till](#)

Arrangör:  [Lägg till](#)

Uthyrd av:  [Lägg till](#)

Temp sal:  (default: 20)

Temp serv:  (default: 21)

☐ Återkommande

☒ Betalt

[Gå tillbaka](#)

Figur 10a, b. Registrera en ny bokning respektive ändra på en sparad bokning

Här kan man gå in och ändra på all information som rör en bokning.

## **6 RESULTAT OCH DISKUSSION**

### **6.1 Resultat**

Resultatet av examensarbetet är en fungerande bokningskalender och datorstyrning av värmesystemet. Bokningskalendern är funktionsmässigt sett färdig, dock kommer utseendet ännu att optimeras. Den har ännu inte lagts ut på Internet då föreningen ännu väntar på tillgång till Multitronics databashanterare. Utvecklingsmiljön är näst intill identisk med deras server, så flytten förväntas inte skapa några problem.

Vidare har skript för att plocka ut bokningarna ur kalendern och föra över dem till justerprogrammet utvecklats. Själva justerprogrammet samt hårdvaran och operativsystemet som programmet körs på har installerats. I skrivande stund är inte arbetet med det elektroniska styrsystemet ännu färdigt, så vi har tvingats skjuta upp testkörningen av systemet.

### **6.2 Fortsatt utveckling**

Hantering av fel som uppstår i värmesystemet och kommunikationen mellan justerprogrammet och det elektroniska styrsystemet är en av de saker som man kunde vidareutveckla och reda ut vilka möjligheter som finns för hur man kan meddela om att felsituationer har uppstått.

Vidare kunde man även effektivera hur värmesystemet styrs genom att analysera data från loggningen och kanske även inkludera en utomhustemperaturmätare i systemet, då uppvärmningen beter sig annorlunda beroende på temperaturen utomhus.

Detta är alla saker som har funderats på men som har prioriterats bort på grund av tidsbrist.

### 6.3 Reflektioner

Det har varit mycket lärorikt att arbeta med detta examensarbete. Jag har fått lära mig mycket om t.ex. 1-Wire, som var en helt ny teknik för mig, och elektroniska komponenter som lätt kan användas tillsammans med en dator. Dessutom har jag fått öva mig på att plocka fram och presentera data ur en databas på ett vettigt och snyggt sätt genom att använda PHP.

Jag har fått lära mig att planera hur man fördelar tiden så att alla delar som ingår i projektet hinns med, samt vikten av att kunna ta tag i de problem man stöter på.

## 7 KÄLLFÖRTECKNING

- /1/ 1-Wire  
<http://en.wikipedia.org/wiki/1-Wire>  
(läst 1.2.2011)
- /2/ Gerby Ungdomsförening r.f:s hemsida  
<http://www.gerby.fi>  
(läst 27.2.2011)
- /3/ Granholm, Richard (2007)  
*Planering av styrsystem för uppvärmning*  
Vasa: Svenska Yrkehögskolan, teknik och kommunikation.
- /4/ MAXIM 1-Wire Tutorial  
<http://www.maxim-ic.com/products/1-wire/flash/overview/index.cfm>  
(läst 1.2.2011)
- /5/ MySQL  
<http://en.wikipedia.org/wiki/MySQL>  
(läst 28.2.2011)
- /6/ MySQL :: About MySQL  
<http://www.mysql.com/about/>  
(läst 8.2.2011)
- /7/ MySQL :: MySQL Database 5.5  
<http://www.mysql.com/products/enterprise/database/>  
(läst 28.2.2011)
- /8/ OpenWrt Wiki  
<http://wiki.openwrt.org/>  
(läst 8.2.2011)
- /9/ OWFS Develompent Site  
<http://owfs.org/index.php?page=what-is-1-wire>  
(läst 1.2.2011)
- /10/ OWSERVER(1) manual page  
<http://owfs.sourceforge.net/owserver.1.html>  
(läst 10.2.2011)

- /11/ PHP: Hypertext Preprocessor  
<http://www.php.net/>  
(läst 8.2.2011)
- /12/ Router  
<http://en.wikipedia.org/wiki/Router>  
(läst 26.3.2011)
- /13/ SQL  
<http://sv.wikipedia.org/wiki/SQL>  
(läst 8.2.2011)
- /14/ SÖU: Gerby Ungdomsförening r.f.  
<http://gerbyuf.sou.fi/start/>  
(läst 2.3.2011)
- /15/ Wikibooks: Lösenordshantering – MD5 och salt  
[http://sv.wikibooks.org/wiki/L%C3%B6senordshantering\\_-\\_MD5\\_och\\_salt](http://sv.wikibooks.org/wiki/L%C3%B6senordshantering_-_MD5_och_salt)  
(läst 6.3.2011)
- /16/ World Wide Web Consortium  
<http://www.w3.org/>  
(läst 1.2.2011)

Beskrivning	Läge 1	Läge 2	Läge 3	Läge 4	Läge 5	Läge 6	Läge 7	Läge 0
	- Datorläge 1, hela auto. - Temp. Inst. från dator	- Datorläge 2, hela auto. - Temp. inst. från panel	- Temp från dator	- Manuellt normalt läge	- Manuellt sänkt temp.	- Förinställt läge	- Man. manuellt läge	- Avstängt
UTG Spjäll sal	- Styrs direkt från dator	- Styrs direkt från dator	- Stängs från inst. värde om sal varmare än serv - Öppnas från inst. värde om sal kallare än serv	- Stängs från inst. värde om sal varmare än serv - Öppnas från inst. värde om sal kallare än serv	- Stängs från inst. värde om sal varmare än serv - Öppnas från inst. värde om sal kallare än serv	- Stängs från inst. värde om sal varmare än serv - Öppnas från inst. värde om sal kallare än serv	- Antar läge som potentiometern anger	
ING Spjäll sal	n/a	n/a	n/a	n/a	n/a	n/a	- Från panelens pot.	
UTG Spjäll serv.	- Styrs direkt från dator	- Styrs direkt från dator	- Stängs om temp. blir 1 grad varmare än inställt	- Stängs om temp. blir 1 grad varmare än inställt	- Stängs om temp. blir 1 grad varmare än inställt	- Stängs om temp. blir 1 grad varmare än inställt	- Antar läge som potentiometern anger	
ING Spjäll serv.	n/a	n/a	n/a	n/a	n/a	n/a	- Från panelens pot.	
UTG Friskluft	- Styrs direkt från dator	- Styrs direkt från dator	- 0% om temp < -1°C annars inst. värde	- 0% om temp < -1°C annars inst. värde	- 0%	- 0%	- Antar läge som potentiometern anger	
ING Friskluft	n/a	n/a	- Från dator	- Från panelen	n/a	n/a	- Från panelens pot.	
UTG Shunt	- Styrs direkt från dator	- Styrs direkt från dator	- Temp< temp-1°C 100% - Temp > temp + 1°C 0% - Panna < 60°C 0% - -1°C<temp<1°C 0-100%	- Temp < temp -1°C 100% - Temp > temp + 1°C 0% - Panna < 60°C 0% - -1°C<temp<1°C 0-100%	- Temp < temp -1°C 100% - Temp > temp + 1°C 0% - Panna < 60°C 0% - -1°C<temp<1°C 0-100%	- Temp < temp -1°C 100% - Temp > temp + 1°C 0% - Panna < 60°C 0% - -1°C<temp<1°C 0-100%	- Antar läge som potentiometern anger	
ING Shunt	n/a	n/a	n/a	n/a	n/a	n/a	- Från panelens pot.	
ING Läge	n/a	n/a	- Från dator	- Från panelen	- Från panelen	- n/a (låst läge)	- Från panelens brytare	
UTG Fläkt	- Styrs direkt från dator	- Styrs direkt från dator	-H0 om panna < 60°C -H0 om temp>temp+1°C -H1 om temp<temp-1°C -Annars H I	-H0 om panna < 60°C -H0 om temp>temp+1°C -H1 om temp<temp-1°C -Annars H I	-H0 om panna < 60°C -H1 om shunt nått 90% tills shunt < 50% -H1 om temp<temp-1°C -Annars HI	- Styrd av brytare på panelen	- Styrd av brytare på panelen	
ING Fläkt	n/a	n/a	n/a	n/a	n/a	- Från panelens brytare	- Från panelens brytare	
ING Temp.	- Styrs direkt från dator - Värdena från bok.kal.	- Styrs direkt från dator - Värdena från pan. pot.	- Styrs direkt från dator - Värdena från bok.kal.	- Från panelens pot. för normal temperatur	- Från panelens pot. för sänkt temperatur	- Trimpt. på kretskortet	- Från panelens pot. för normal temperatur	
Mätn Temp.	- Mäts med DS-givare	- Mäts med DS-givare	- Mäts med NTC-givare	- Mäts med NTC-givare	- Mäts med NTC-givare	- Mäts med NTC-givare	- Mäts med NTC-givare	